Verde Users Manual Version 2.5

Karl G. Merkley, Ray J. Meyers

December 14, 2001

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Verde (Verification of Discrete Elements) is a program designed to verify the properties of a finite element mesh stored in Exodus II format[1][2]. Verde uses a variety of verification algorithms and graphics to thoroughly analyze the individual and combined properties of the mesh and to help the user detect any problems that may exist. Verde can be run in two modes. The default mode provides a graphical user interface (GUI) and tools for visualizing the model and displaying problem areas of the mesh. The GUI was developed using Qt, the cross-platform C++ GUI toolkit [3]. Verde can also be run from the command line in a batch mode. It is designed for high throughput of large meshes. It is easily extensible and built to allow the addition of new element types and metrics as they become available.

1.1 Element Types

Verde will analyze the following element types:

Hexes	Tetrahedra
Pyramids	Wedges (triangular prisms)
Knives	Quads
Triangles	Bars (beams)

Verde will handle all higher-order variants of these types provided they are defined using the standard Exodus II node numbering; however, for higher-order elements, the metrics are calculated based on the linear versions of the elements.

1.2 Types of Verification

Verde 2.5 performs mesh verification in three areas:

1. *Individual Element Metrics.* The individual characteristics of each element are verified by calculating common metrics depending on the element type. Verde calculates one or more metrics for each element type supported. For each metric, the minimum, maximum, average, and standard deviation of each metric are tracked, as well as the elements at which the minimum and maximum occur. All elements whose metric values fall outside the user-definable acceptable range are flagged as failed.

2. *Topology Checks.* The topology of the mesh is carefully analyzed, looking for any defects in the mesh connectivity or continuity that would invalidate the mesh. Verde reports the number of exterior entities for the mesh, checks for manifold topology, and if the object is manifold, calculates the Euler number and infers the probable overall topology. In addition, it checks for adjoining quadrilateral faces in the mesh that share only three nodes.

3. *Interface Checks.* The exterior surface of the mesh is carefully analyzed. Exterior elements are checked for coincidence. In this way, potential problems such as incorrect mesh joins and multiply-meshed regions can be detected. In batch mode, an optional output file can be created (in Exodus II format) which contains information from the analysis such as the exterior skin of the mesh, inferred model edges, and failed elements. This file can be read by Verde in the interactive mode or other tools that read Exodus II data to establish orientations and locations of bad elements and to explore and understand how to improve the current mesh.

1.3 Invoking Verde

The Verde graphical user interface (GUI) can be invoked from the command prompt.

% verde [options] [journal_file.jou] [input_file.gen]

where the input file is the file to be verified (in Exodus II format) and the options are optional control flags. Options are preceded by the "-" character ("minus" or "dash") and can be specified in any order. Current valid options for version 2.5 are:

-help	Print this summary
-batch	Runs verde in batch mode
-blocks <$val>	Specifies block(s) to be processed (e.g. 1,5-7,9)
-individual	Prints metrics for each block individually
-list_defaults	Lists defaults settable in .verde defaults file
-defaults <$val>	Specifies path and file of defaults file (e.g ../defaults1)
-nodefaults	Do not process .verde default file
-nointerface	Suppresses mesh interface verification calculations
-nometrics	Suppresses element metric calculations
-output <$val>	Generates an Exodus II output file specified by *val*
-notopology	Suppresses mesh topology verification calculations
-print_failed_elements	Prints individual failed elements and values
-version	Prints the verde version number
-warning_level <$val>	A value less than 2 will suppress the warning for quadrilateral elements that share three nodes

If Verde is invoked without the optional file name, a blank screen is displayed and the user may select a file using the File/Open menu item.

The basic syntax of the Verde batch command is:

% verde –batch [options] [journal_file.jou] input_file.gen

For more details regarding the batch execution of Verde, see section 3, Controlling Verde in Batch Mode.

Chapter 2

An Overview of the GUI

The Verde GUI is composed of a menu bar, a tool bar, three dockable windows, and a graphics display area as shown in figure 2.1.

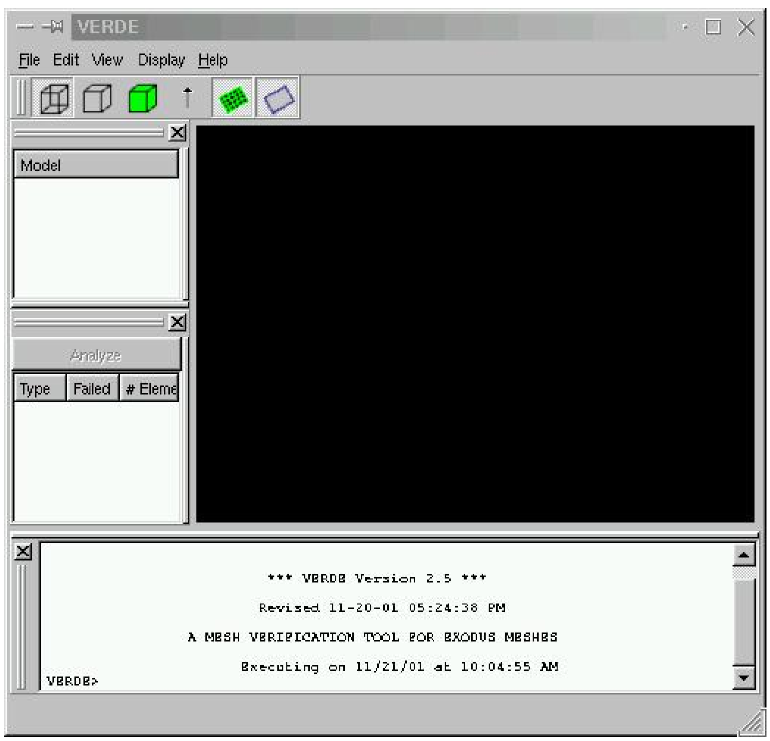

Figure 2.1: Verde Main Screen.

The menu bar includes functions for File operations such as loading a model, printing, saving an image file and exiting the program. The Edit menu allows the user to specify parameters for the element metrics and user preferences. The View menu provides controls the visibility of the dockable windows. The Display menu allows the user to control the graphics display of the model. The Help menu provides access to this documentation.

The tool bar allows the user to specify the graphics display modes of the model as wire frame, hidden line, or shaded image. It also provides an option to display the normals of a model.

There are three dockable windows. The Model window controls the portions of the model that are displayed. The Results window controls the display of verification results, and the Command line window allows command input and is used to echo textual results.

2.1 Controlling the Verde GUI

The following sections will give an explanation of each of the windows, menu items and dialog boxes in the Verde GUI.

2.1.1 Model Window

The Model Window is located in the upper left hand corner by default. An example of this window is shown in figure 2.2. This window shows the current model that is loaded and a list of the blocks, node sets and side sets that are active and inactive in the model. Right clicking on any element at the lowest level of the tree structure will bring up a pop up menu that provides options for activating or deactivating any given item. Multiple items may be selected using the shift or control keys. When a block is inactive, the data for that block is not in memory. It will not be displayed and no calculations will be performed on that block. When a block is made active, the data for that block is read from the Exodus II file, the block is displayed and calculations can be performed. Node set and side set activation is independent of block activation.

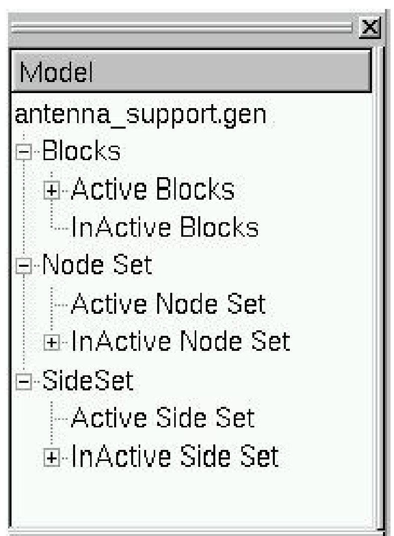

Figure 2.2: Verde Main Screen.

Node Set Display

Node sets are displayed as filled magenta squares drawn at each node where the node set is applied as shown in figure 2.3. This display is useful for verifying the correct application of nodal boundary conditions in the model.

Side Set Display

Side sets are displayed as magenta facets drawn on the face or edge of each element where the side set is applied as shown in figure 2.4. The side set display facets are slightly smaller than the faces of the elements

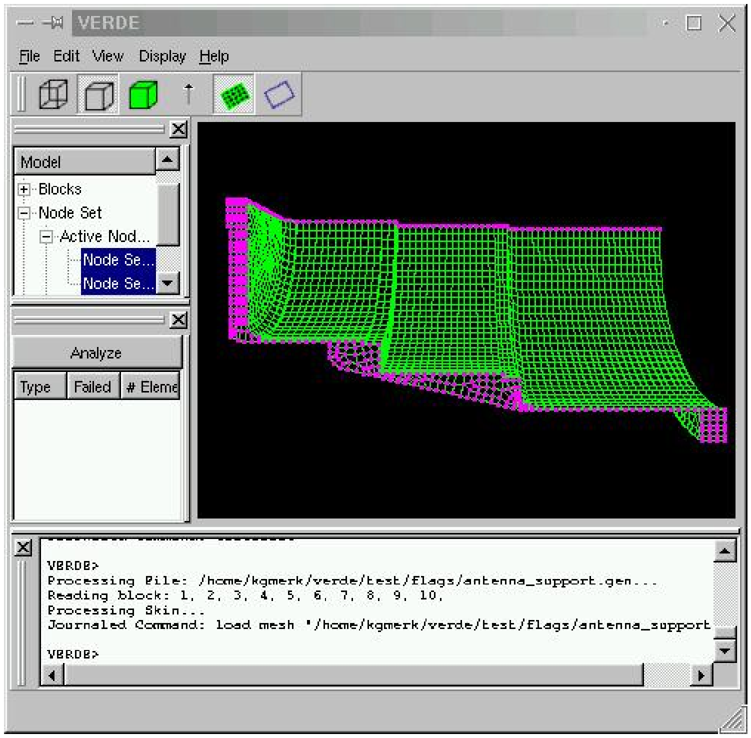

Figure 2.3: Node Set Display.

that they are drawn on. This allows the user to verify that individual elements were not skipped when the boundary condition was applied.

Side sets defined on the face of an element are only visible when viewed from the side that they are defined on. For example, shell elements may have side sets placed on the forward or the reverse face. A side set placed on the forward face will not be visible when viewed from the reverse side of the model.

2.1.2 Results Window

The Results Window is typically located below the Model Window. The Analyze button is located on this window and is inactive, as shown by its grayed out status, until a model is imported into Verde. After a successful analysis of the model, a list of element types, failure cases (metric, topological, and interface), and number of failed components is displayed in this window. Selecting one of these rows causes the failed items to be highlighted on the graphics model. The results of a metrics calculation is shown in figure 2.5.

2.1.3 Command Line Window

The Command Line window is normally located along the bottom of the main Verde window. This window allows the user to type in Verde commands. It is also used to display the textual results of the calculate command as shown in figure 2.6.

It is not possible to execute all commands possible in the GUI from the command line. For example, there are no command line options to rotate, translate or zoom the model. However, most commands are available in both modes. A complete list of the command line options is given in Appendix C.

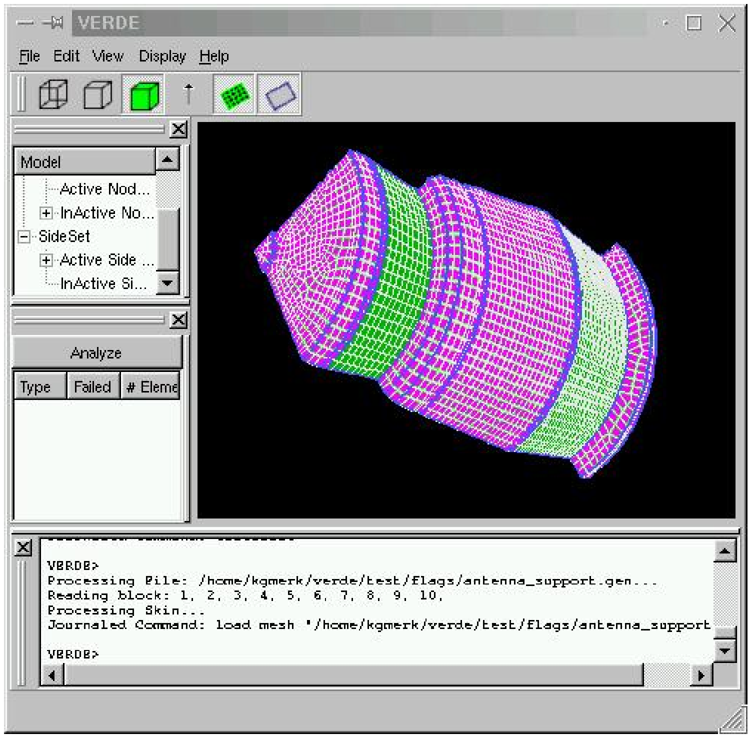

Figure 2.4: Side Set Display.

2.1.4 File Menu

The File menu provides standard file manipulation capabilities for loading an Exodus II file into Verde, reading and recording journal files, printing, capturing screen images, and exiting the program.

Open Mesh

If Verde is invoked without the optional file name, a blank screen is displayed and the user must select a model using the File/Open Mesh dialog as shown in figure 2.7.

The Open Mesh dialog includes an option that allows the user to select the blocks that will be imported from the Exodus II file. The block selection dialog lists the blocks that are defined in the Exodus II file and the element type for each block. Verde considers the imported blocks “active”. Only active blocks are displayed in the graphics region and considered during metrics and topology calculations.

When the mesh has been imported the skin of the model is displayed in the viewing region as shown in figure 2.8.

Only one file may be opened at a time in Verde 2.5. If a second mesh file is opened the data from the first mesh is deleted.

Play Journal File

The Play Journal File option brings up a file open dialog that allows the user to select a journal file and replay a set of predefined Verde commands that are stored in that file.

Appendix C gives a description of the commands used in a journal file.

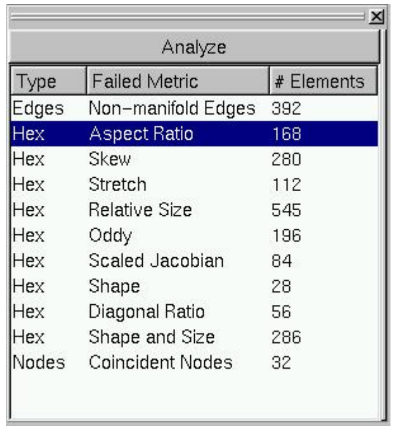

Figure 2.5: Results Window.

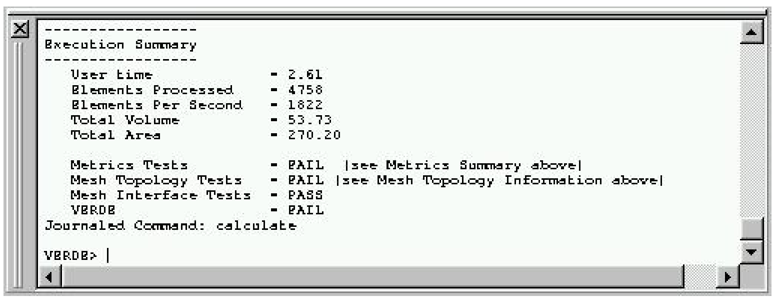

Figure 2.6: Command Line Window.

Record Journal File

The Record Journal File option brings up a file save dialog that allows the user to specify a file to save the commands used during a Verde session. As soon as this file is opened, all journal commands are written into this file. Note that most graphics manipulation commands are not supported in journal files in Verde version 2.5.

End Record Journal File

This option stops the journaling process. No additional commands are written to the specified journal file.

Save Image As

This menu item allows the user to save the currently displayed view as .PNG image file. A .png extension will be appended to the file name if the extension doesn't already exist. This image format can be imported into many display and manipulation programs including the Microsoft Office Suite^{TM}.

Print

The print option brings up a print dialog and allows the user to print the currently displayed image or save the print image to a file. This method currently prints a raster display of the image, so some degradation of image quality may occur due to the printer resolution.

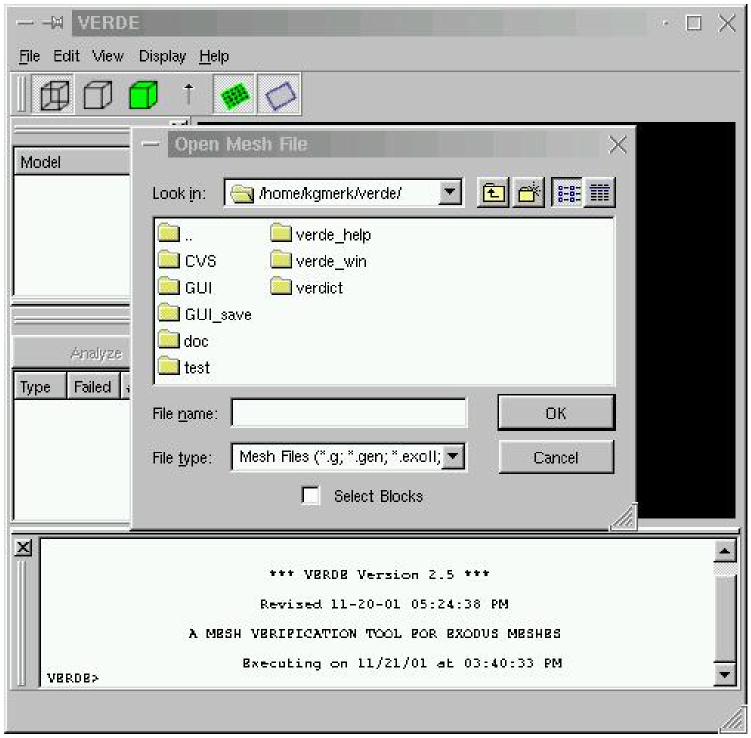

Figure 2.7: Verde Open Mesh dialog.

If the image is displayed on the default black background, a message box is displayed that allows the selection of a white background. If the user clicks yes in this message box the display region will momentarily flash with a white background. The image will then be printed in the current line color and a white background.

Note: Verde 2.5 does not currently provide a method of printing the results of metrics or topology calculation. To print this data, the user must select and cut the desired information from the Command Line window and paste it into a separate application (the editor or word processor of choice) for printing.

Exit

Exits Verde. No state or data is saved.

2.1.5 Edit Menu

The Edit menu allows users to control and save parameters that are used by Verde during mesh verification sessions.

Metric Ranges

Pass/fail criteria for the element metrics are based on the ranges that are defined for each metric. The minimum and maximum acceptable range for each element and metric can be defined by invoking the Metric Ranges dialog. This dialog box is shown in figure 2.9. The Metric Ranges dialog contains tabs for each element type that is supported by Verde. Each tab contains fields for the minimum and maximum acceptable range for each metric. Clicking either the OK or Apply buttons activates the changed values. The

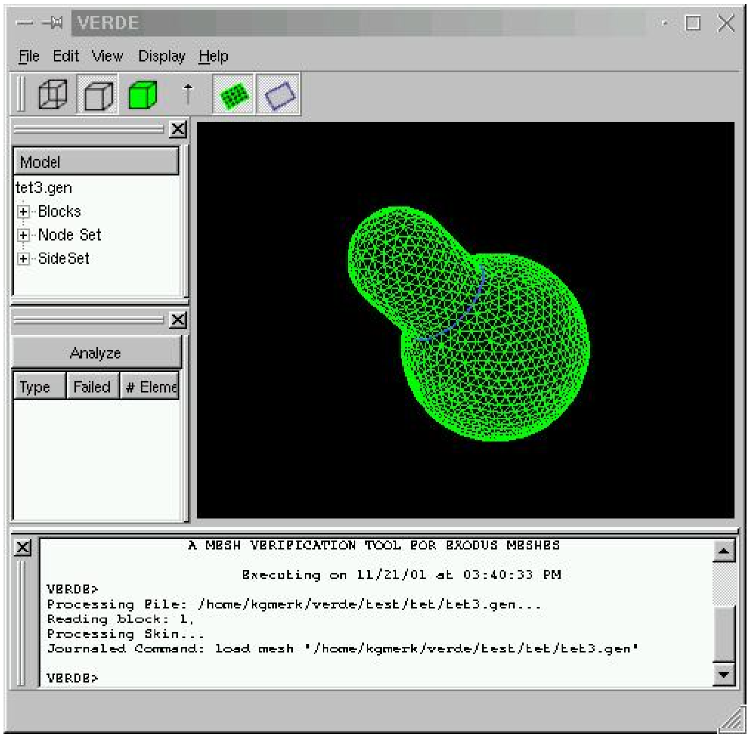

Figure 2.8: Model skin displayed in hidden line mode.

Restore button restores the values to the default state. Verde 2.5 does not yet support interactively writing or reading values from the .verde file. However, this file is read at start up and the values are displayed in the Metrics Ranges dialog and used for metrics calculations.

Preferences

The Preferences dialog box is shown in figure 2.10. This dialog box can be used to specify the types of calculations the Verde will perform, the severity of warning levels that are displayed, the data that should be echoed to the Command Window, and the user's choice of metrics ranges files.

2.1.6 Display Menu

The Display Menu controls how the graphics are displayed in the display region. The display region is drawn using the portable OpenGL^{TM} graphics API.

Wireframe/Hidden/Smooth Shade

These toggle actions control how the model is displayed. The graphics modes can be controlled from both the Display menu and the Verde toolbar.

Normals

This menu item toggles the display of normal vectors on the model. Figure 4.4 shows an example of this type of display.

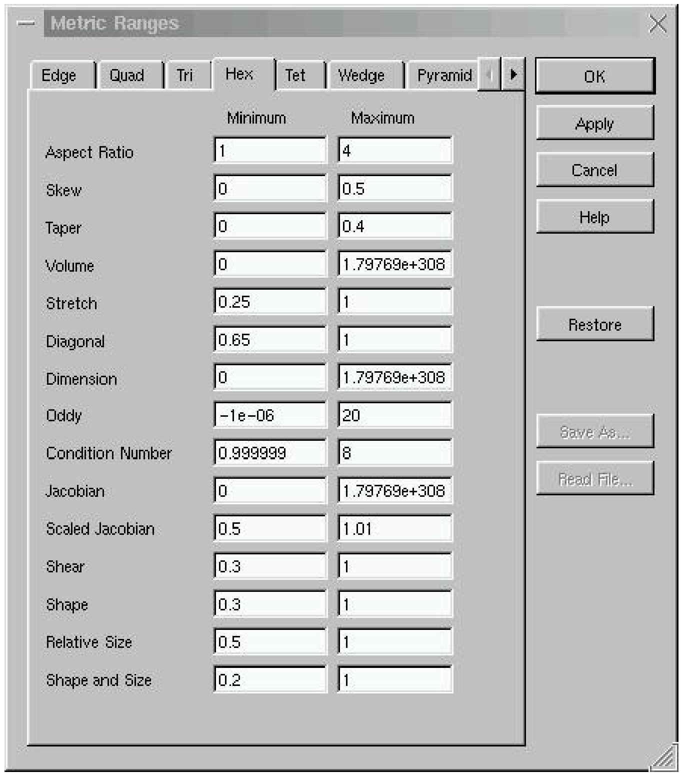

Figure 2.9: Metric Ranges dialog box.

Skin

This menu item toggles the skin on or off. Turning off the skin can be useful for visualizing failed elements or interfaces internal to the model.

Model Edges

This menu item toggles the model edges on or off. Turning off the skin and turning on the model edges is useful for visualizing failed elements or interfaces internal to the model.

Verde displays both inferred edges and free edges, edges which belong to only one polygon. The inferred model edges are determined by calculating feature angles between adjacent polygons. This can cause problems if the object is primarily smooth. For example the object displayed in figure 2.8 only has one edge as shown in figure 2.11

Foreground/Background Color

The display menu also provides methods of setting the foreground and background colors of the model via a color dialog box as shown in figure 2.12.

The background and foreground can be set to any color by choosing either one of the predefined colors in the color table or by creating a custom color. Note that in version 2.5 of Verde the custom colors are not maintained between sessions.

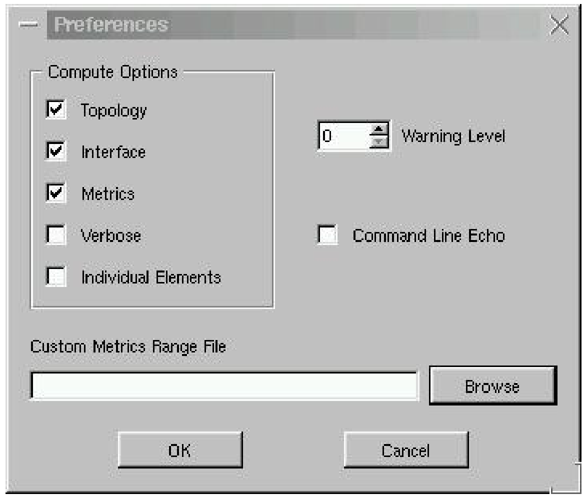

Figure 2.10: Preferences dialog box.

Reset

This function returns the graphics model to its default rotation, scale, and translation, such that the model center is located at the center of the display region.

Model Manipulation

In addition to the display options that are available from the Display menu, the graphics model may also be transformed using the mouse.

Left mouse:	Rotation about the model X and Y axes.
Right mouse:	Translation in the screen X and Y axes.
Middle mouse:	Zoom in the screen Z axis.

2.1.7 Help

Displays this document.

The help system should be configured properly by the install process. However, if an error should occur check the following items.

1. The environmental variable VERDE_DIR should point to the directory where the verde executable is located. The documentation (verde*.html) is located in a sub-directory of VERDE_DIR named *doc* ($VERDE_DIR/doc).

2. The helper application *verde_help* should exist in the VERDE_DIR directory.

If either of these items is incorrect, you can correct the problem manually or attempt to reinstall verde.

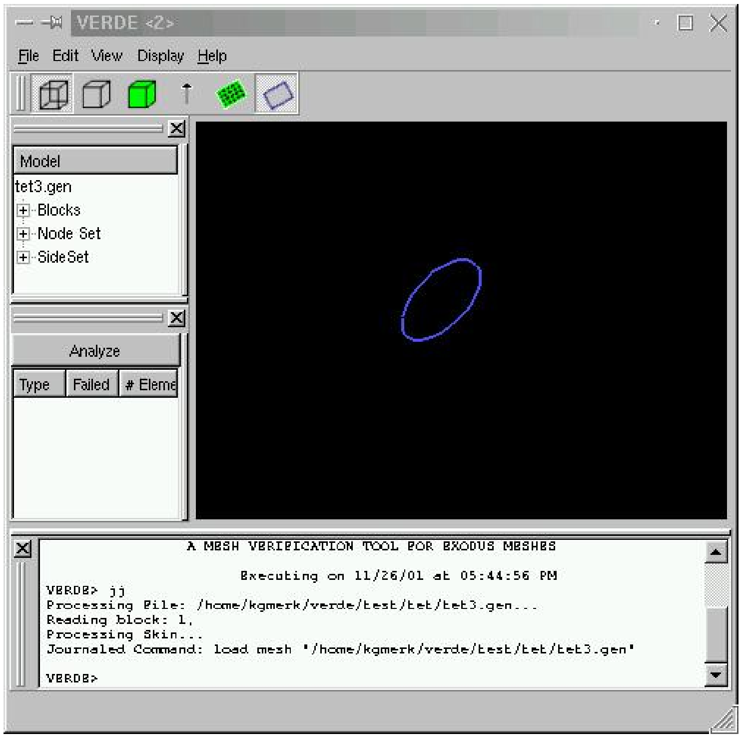

Figure 2.11: Edge calculated for a relatively smooth object.

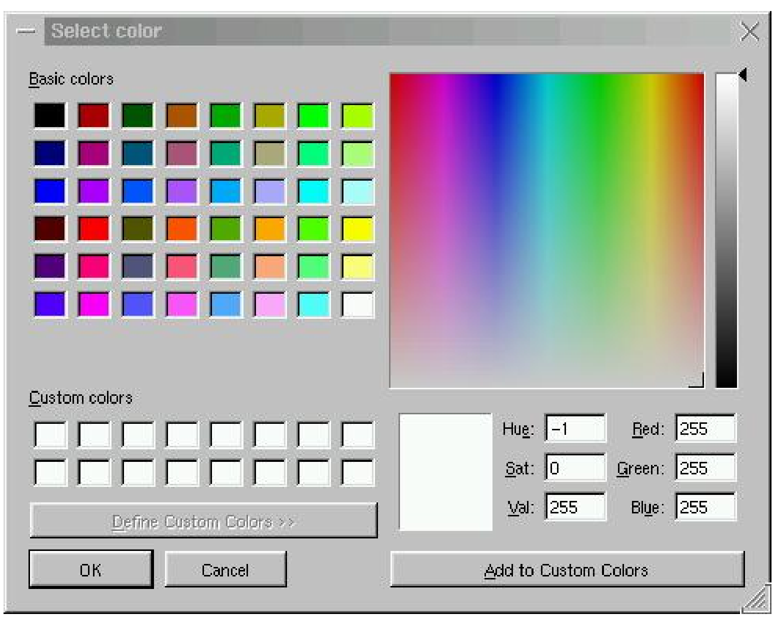

Figure 2.12: Select Color dialog for foreground and background colors.

Chapter 3

Controlling Verde in Batch Mode

Verde can be run as a batch-mode program when invoked from the command prompt with the "–batch" option.

The basic syntax of the Verde batch command is: [1]

% verde –batch [options] input_file.gen

where input_file is the file to be verified (in Exodus II format) and the options are optional control flags. Options are preceded by the "-" character ("minus" or "dash") and can be specified in any order. Current valid options for version 2.5 are:

-version	Prints the Verde version number
-help	Print this summary
-batch	Runs verde in batch mode
-blocks <$val>	Specifies block(s) to be processed (e.g. 1,5-7,9)
-individual	prints metrics for each block individually
-list_defaults	Lists defaults settable in .verde defaults file
-defaults <$val>	Specifies path and file of defaults file (e.g ../defaults1)
-nodefaults	do not process .verde default file
-nointerface	Suppresses mesh interface verification calculations
-nometrics	Suppresses element metric calculations
-output <$val>	Generates an Exodus II output file specified by *val*
-notopology	Suppresses mesh topology verification calculations
-print_failed_elements	Prints individual failed elements and values
-version	Prints the verde version number
-warning_level <$val>	A value less than 2 will suppress the warning for quadrilateral elements that share three nodes

Used without any options, e.g.,

% verde –batch test_file.gen

Verde will perform all available verification procedures to the file test_file.gen. The most common options are specified to suppress one or more verification procedures on the file. For large files, the savings in execution time resulting from suppressing verification procedures can become important.

[1] Under MS Windows^{TM} operating systems users should run verde, as documented, and **not** verde.exe. The batch version is run from verde.com. This file provides console output capabilities (such as redirection and pipe) similar to the UN*X console mode. The verde.com file resides in the same directory as the verde executable.

Chapter 4

Program Output

When Verde is used to verify a mesh, by default it sends output to the command line and optionally (using -output) to an auxiliary graphics output file. Option flags control the exact output at the command line, and can also be used to disable the graphics file output.

4.1 GUI Output

The graphical user interface provides a method of visualizing the results from a Verde analysis. After clicking on the Analyze button, failure cases are listed in the Results window. If nothing is listed, the model passed all tests for the currently defined set of metric ranges. The results from the graphical user interface is primarily based the visual display of the results data. The results to be displayed are selected by clicking on one of the items in the Results window. If there are no items displayed the model passes all requested tests.

By default, element metric, topological, and interface analyses are performed on the model. Specific analyses may be excluded by deselecting them in the Edit/Preferences dialog box.

The following sections outline the types of results that are available for each analysis type.

4.1.1 Individual Element Metrics

. The results for individual element metrics can be displayed by clicking on the appropriate entry in the Results window. An example of the resulting display is shown in figure 4.1. This display was created by turning model shell off and rendering the failed elements in hidden line mode. This provides a useful method of viewing bad elements internal to the model.

4.1.2 Topology Checks

A topological analysis is performed that determines if the model is manifold, composed of complete solids or shells, or non-manifold. Non-manifold objects may include a mixture of bar, shell and/or solid elements, or they may be objects that are not correctly closed. Non-manifold edges may indicate the presence of hinges in the model.

The command window prints a prediction of the probable topology of the model. For example, the topology information for figure 4.4 is calculated as:

```
-----------------------------
Mesh Topology Information
-----------------------------
   Probable Topology:  Open Surface with 11 hole(s)
   Possible Topologies: N Open surfaces with N+10 hole(s)
```

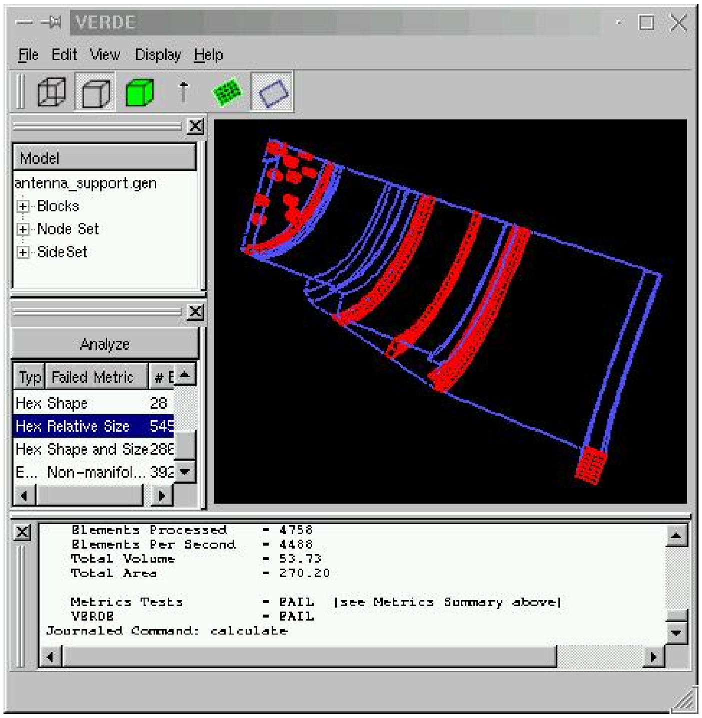

Figure 4.1: Sample results display of failed elements.

This calculation is based on the Euler-Descartes formula $E = v - e + f$, where v is the number of vertices, e is the number of edges, and f is the number of faces[4].

Verde 2.5 can display non-manifold edges as shown in figure 4.2. A non-manifold edge occurs when more than two entities share a boundary edge. The figure shows two blocks that are only connected along one edge. This is not a valid solid object. If the model is not properly constrained, it could result in a hinge that may cause some analysis codes to fail.

Figure 4.3 demonstrates another case that reports non-manifold edges in Verde 2.5. In this case there are shell elements superimposed on a set of hexahedral elements. Once again, this may be correct, but it allows the user to verify that the desired elements are in the correct location. Bar elements which share an edge with other elements will also be flagged as non-manifold entities.

The topology of a model can also be checked by displaying the element normals. Figure 4.4 shows a shell model with the normals turned on. Note that in the upper left hand corner the normals of the flange have the opposite sense of the normals on the rest of the model. Inverted normals can cause some analysis programs to calculate negative areas for some of the elements and may result in an aborted analysis.

4.1.3 Interface Checks

Verde 2.5 also provides tools to verify the interfaces between sections of a model. Figure 4.5 shows red squares at locations where two nodes are coincident. This may not necessarily be an error in the model if the nodes are in sliding contact or are constrained by some type of multi-point constraint.

Verde also detects coincident quadrilaterals and triangles. This may indicate that the model is not correctly joined.

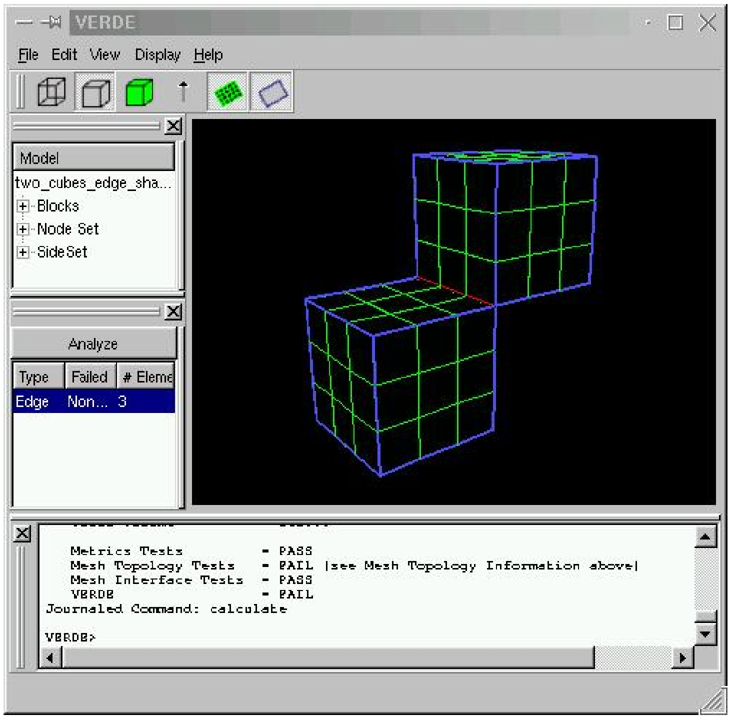

Figure 4.2: Display of non-manifold edges.

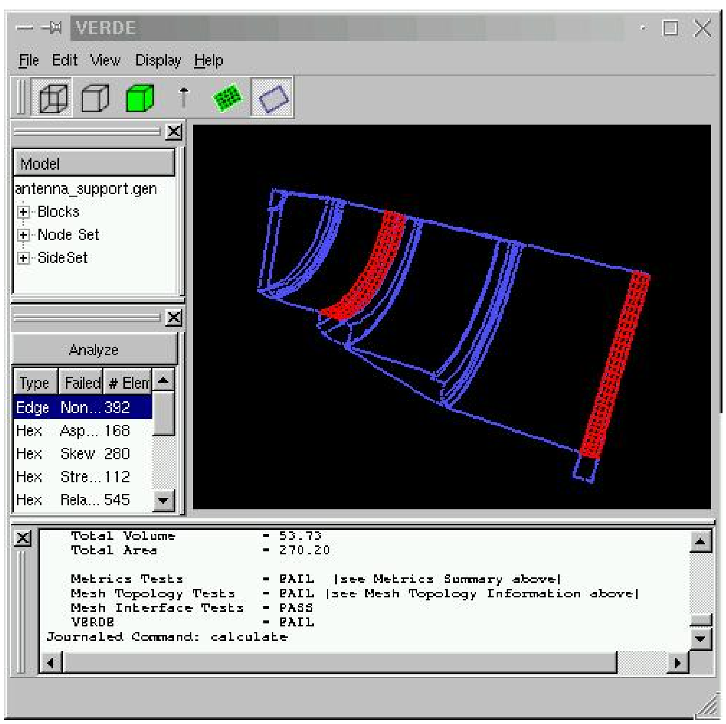

Figure 4.3: Display of shell elements superimposed on hex elements.

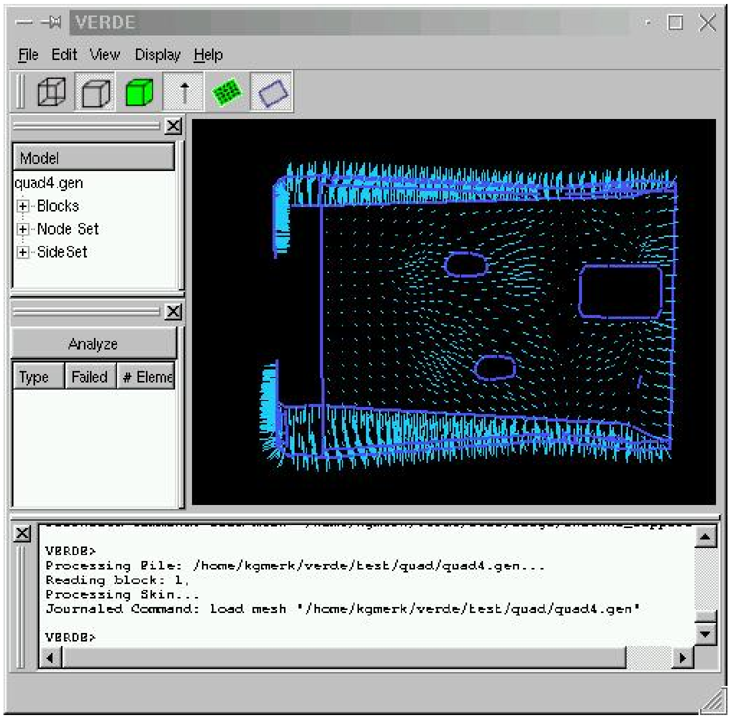

Figure 4.4: Display of element normals on a shell model.

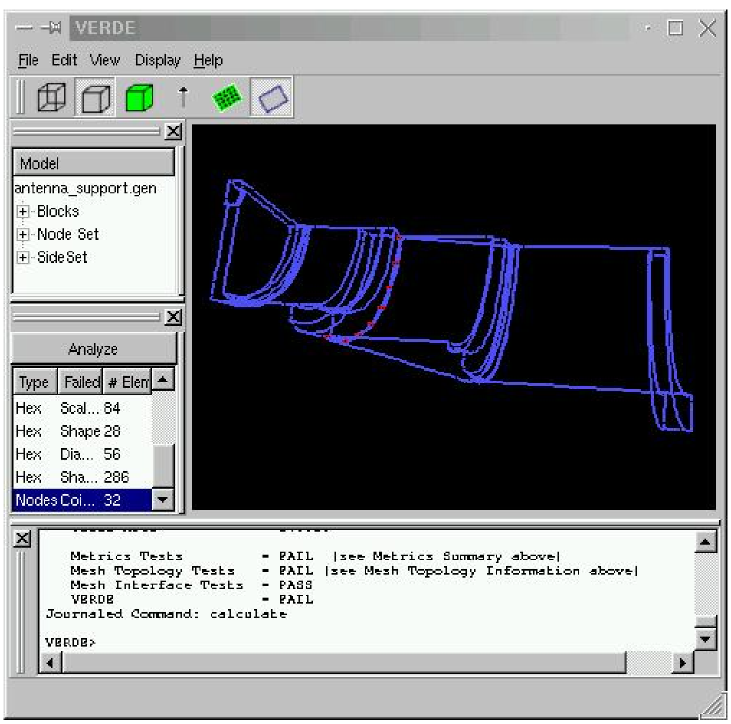

Figure 4.5: Display of coincident nodes in model.

4.2 Command Line Output

Command line output is divided into seven sections:

1. Banner: This prints the version of Verde being run, the revision date of Verde, and the execution date of the run.
2. Element Metrics: In this section, metric information is output for each element type in the input file. If the -individual option is used, metric information is output for each individual element block.
3. Metric Summary: This section lists information about the number of elements of each type that fail the various metric tests. If the -verbose option flag is specified, each element is listed, along with the failed metric value for the element.
4. Mesh Topology Information: This section lists any possible topology problems found in the mesh, and states whether the mesh appears to be conformal or not. The probable topology of the mesh model is also listed based on the Euler number calculated for this body.
5. Mesh Interface Information: This section gives summary information about the exterior skin of the model and the inferred model edges found. If any potential problems with the exterior are discovered (such as incomplete or incorrect mesh joining, non-conformal interfaces, etc) these are listed as warnings.
6. Output Information: This section lists the information that is written to the graphics output file when the -output option is invoked. The file is written in Exodus II format.
7. Execution Summary: This section lists summary information about the size and execution speed of the run, and gives a summary of each verification procedure, listing the result as PASS, FAIL, or WARN, as well as an overall PASS/FAIL for the Verde run. If any FAIL or WARN occurs in any verification procedure, the overall Verde run is flagged as FAIL. NOTE: A failure condition in Verde does not always mean the mesh is incorrect. For example, a mesh with contact surfaces would generate a WARN in the Mesh Interface Information section, but would be a valid mesh for the problem.

Listed below is the output for a sample run of Verde:

```
% verde -batch -output output_file.gen antenna_support.gen


       *** VERDE Version 2.5 ***

      Revised 12-12-01 08:10:44 AM

VERDE: VERIFICATION OF DISCRETE ELEMENTS

A MESH VERIFICATION TOOL FOR EXODUS MESHES

      Executing on 12/12/01 at 02:04:41 PM

This version of Verde is only for use inside Sandia

Executing: verde -batch -output output_file.gen antenna_support.gen

Processing File: test/flags/antenna_support.gen...
Reading block: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
```

```
Processing Skin...

---------------
Element Metrics
---------------
Processing block 1, 261 elements of type HEX:

Processing block 2, 84 elements of type SHELL:

Processing block 3, 616 elements of type SHELL:

Processing block 4, 56 elements of type SHELL:

Processing block 5, 1820 elements of type HEX:

Processing block 6, 588 elements of type HEX:

Processing block 7, 84 elements of type SHELL:

Processing block 8, 189 elements of type SHELL:

Processing block 9, 21 elements of type SHELL:

Processing block 10, 1039 elements of type HEX:


Summary Hex Metrics, total = 3708 Hexes:

   Function Name     Average    Std Dev    Minimum   (id)  Maximum  (id)
 -----------------  ---------  ---------  --------------  -------------
      Aspect Ratio  2.0273290   0.875322  1.0112389 (2979)  5.3614430 (398)
              Skew  0.1713917   0.181553  0.0000930 (3106)  0.7829728 (421)
             Taper  0.0090899   0.009527  0.0000000 (1)  0.0346601 (2857)
    Element Volume  0.0027304   0.001684  0.0004244 (3654)  0.0093638 (57)
           Stretch  0.6235547   0.165799  0.1858189 (300)  0.8732918 (3665)
    Diagonal Ratio  0.9264883   0.073955  0.6338562 (601)  0.9999243 (3106)
         Dimension  0.0707981   0.018386  0.0325387 (327)  0.1062584 (57)
              Oddy  4.4684236   9.235309  0.1271596 (2978)  52.721605 (401)
     Condition No.  1.4180944   0.518734  1.0209791 (2935)  3.9543516 (299)
          Jacobian  0.0022765   0.001574  0.0003001 (3654)  0.0086663 (57)
   Scaled Jacobian  0.9003169   0.132059  0.4421353 (391)  0.9996536 (3113)
             Shear  0.9238993   0.108117  0.5168335 (391)  0.9997690 (3113)
             Shape  0.7674397   0.182423  0.2528860 (299)  0.9794520 (2935)
     Relative Size  0.6983444   0.173235  0.1998347 (4)  0.9737509 (3059)
    Shape and Size  0.5507035   0.197470  0.0749352 (333)  0.8845074 (3059)

Summary Quad Metrics, total = 1050 Quads:

   Function Name     Average    Std Dev    Minimum   (id)  Maximum  (id)
 -----------------  ---------  ---------  --------------  -------------
      Aspect Ratio  1.5180575   0.179408  1.3196892 (4004)  2.1541519 (738)
              Skew  0.0000000   0.000000  0.0000000 (88)  0.0000000 (88)
             Taper  0.0000000   0.000000  0.0000000 (88)  0.0000000 (88)
           Warpage  0.0000000   0.000000  0.0000000 (88)  0.0000000 (88)
      Element Area  0.0299522   0.002293  0.0222158 (738)  0.0313723 (6820)
```

```
           Stretch  0.7817586   0.057237  0.5954718 (738)  0.8541115 (4004)
     Maximum Angle  90.000000   0.000006  90.000000 (88)  90.000000 (88)
     Minimum Angle  90.000000   0.000006  90.000000 (88)  90.000000 (88)
              Oddy  0.3926366   0.268801  0.1578855 (4004)  1.4279352 (738)
     Condition No.  1.0922582   0.057388  1.0387217 (4004)  1.3091859 (738)
          Jacobian  0.0299522   0.002293  0.0222158 (738)  0.0313723 (6820)
   Scaled Jacobian  1.0000000   0.000000  1.0000000 (88)  1.0000000 (88)
             Shear  1.0000000   0.000000  1.0000000 (88)  1.0000000 (88)
             Shape  0.9177645   0.042640  0.7638335 (738)  0.9627218 (4004)
     Relative Size  1.0000000   0.000000  1.0000000 (88)  1.0000000 (88)
    Shape and Size  0.9177645   0.042640  0.7638335 (738)  0.9627218 (4004)

---------------
Metrics Summary
---------------
   Number of Failed Hex Metrics: 1755
      Failed Aspect Ratio: 168   (valid range:      1.000 to      4.000)
      Failed Skew: 280   (valid range:      0.000 to      0.500)
      Failed Stretch: 112   (valid range:      0.250 to      1.000)
      Failed Diagonal Ratio: 56   (valid range:      0.650 to      1.000)
      Failed Oddy: 196   (valid range:     -0.000 to     20.000)
      Failed Scaled Jacobian: 84   (valid range:      0.500 to      1.010)
      Failed Shape: 28   (valid range:      0.300 to      1.000)
      Failed Relative Size: 545   (valid range:      0.500 to      1.000)
      Failed Shape and Size: 286   (valid range:      0.200 to      1.000)
   Number of Failed Quad Metrics: 0

----------------------------
Mesh Topology Information
----------------------------
   WARNING: inconsistent edge to quad/tri count in exterior mesh...
   WARNING: inconsistent quad to element count in mesh...

   Mesh is not fully conformal

   Topology: Non-Manifold
   Number Non-Manifold Edges:  598

-------------------------
Mesh Interface Information
-------------------------
   Number Surface Quads:     4448
   Number Surface Tris:      0
   Number Coincident Nodes:  32
   Number Coincident Quads:   0
   Number Coincident Tris:   0

   Number Model Edges:       1260


------------------
Output Information
------------------
   Writing Output to exodus file output_file.gen:
   Writing 4830 exterior nodes...
```

```
   Writing 4448 exterior quads to block 1...
   Writing 1260 model edges to block 2...
   Writing 168 failed hexes (Aspect Ratio) to block 3...
   Writing 280 failed hexes (Skew) to block 4...
   Writing 112 failed hexes (Stretch) to block 5...
   Writing 56 failed hexes (Diagonal Ratio) to block 6...
   Writing 196 failed hexes (Oddy) to block 7...
   Writing 84 failed hexes (Scaled Jacobian) to block 8...
   Writing 28 failed hexes (Shape) to block 9...
   Writing 545 failed hexes (Relative Size) to block 10...
   Writing 286 failed hexes (Shape and Size) to block 11...

-----------------
Execution Summary
-----------------
   User time            = 1.53
   Elements Processed   = 4758
   Elements Per Second  = 3109
   Total Volume         = 10.12
   Total Area           = 31.45

   Metrics Tests        = FAIL  (see Metrics Summary above)
   Mesh Topology Tests  = FAIL (see Mesh Topology Information above)
   Mesh Interface Tests = PASS
   VERDE                = FAIL
```

Appendix A

The .verde Preferences File

The .verde preferences file [1] can be used to customize the behavior of Verde to a large extent. Verde looks for a valid verification file first in the current working directory (./.verde) and then in the home directory ($(HOME)/.verde). If a valid file is found, it is processed first, before any flags on the command line. Therefore, if an option is listed in both the .verde file and on the command line, the command line flag will take precedence. The .verde file can set the same flags as are available on the command line. It is also possible to control the valid range of any Verde metric using the file. Thus, if there are certain element metric limits for a given application, these can be stored in the .verde file and used in place of the default limits.

The file is arranged in a simple token-value format. Listed first on the line is a single word token describing the token to be set. It is followed on the same line by a valid value. A space is used to separate the token and its value. Values can be:

1. Real: Used to represent a real number.
2. Integer: Used to represent an integer.
3. Logical: Used to indicate on/off. Valid values of the logical include: ON, on, OFF, off, TRUE, true, FALSE, and false.

If a line begins with the character '#' it is interpreted as a comment line. Blank lines are ignored. A simple preferences file might look like this:

% cat ./.verde
# Sample .verde preferences file
PERFORM_INTERFACE_CALCULATIONS OFF
HEX_METRIC_MIN_NORM_JACOBIAN 0.15

This file turns off mesh interface calculations (identical to the -nointerface flag on the command line), and sets the minimum normalized jacobian for hex elements to 0.15 (any value less than this will trigger a failed hex element). A full list of valid tokens for setting preferences in Verde can be found in Appendix A. This list can also be obtained on-line by typing:

% verde -list_options

A.1 Valid Tokens for the .verde Preferences File

Available settings in .verde defaults file:

[1] In a future release of Verde the preferences file format documented here will be deprecated and will be replaced by the journal file format.

```
# Command Line Flags:
PERFORM_TOPOLOGY_CALCULATIONS     on/off
PERFORM_INTERFACE_CALCULATIONS    on/off
PERFORM_METRIC_CALCULATIONS       on/off
PERFORM_FILE_OUTPUT               on/off
PERFORM_VERBOSE_OUTPUT            on/off
PERFORM_INDIVIDUAL_OUTPUT         on/off

# Hex Metrics failure criteria:
HEX_METRIC_MIN_ASPECT             real_value
HEX_METRIC_MAX_ASPECT             real_value
HEX_METRIC_MIN_SKEW               real_value
HEX_METRIC_MAX_SKEW               real_value
HEX_METRIC_MIN_TAPER              real_value
HEX_METRIC_MAX_TAPER              real_value
HEX_METRIC_MIN_VOLUME             real_value
HEX_METRIC_MAX_VOLUME             real_value
HEX_METRIC_MIN_STRETCH            real_value
HEX_METRIC_MAX_STRETCH            real_value
HEX_METRIC_MIN_DIAGONAL           real_value
HEX_METRIC_MAX_DIAGONAL           real_value
HEX_METRIC_MIN_DIMENSION          real_value
HEX_METRIC_MAX_DIMENSION          real_value
HEX_METRIC_MIN_ODDY               real_value
HEX_METRIC_MAX_ODDY               real_value
HEX_METRIC_MIN_CONDITION          real_value
HEX_METRIC_MAX_CONDITION          real_value
HEX_METRIC_MIN_JACOBIAN           real_value
HEX_METRIC_MAX_JACOBIAN           real_value
HEX_METRIC_MIN_NORM_JACOBIAN      real_value
HEX_METRIC_MAX_NORM_JACOBIAN      real_value
HEX_METRIC_MIN_SHEAR              real_value
HEX_METRIC_MAX_SHEAR              real_value
HEX_METRIC_MIN_SHAPE              real_value
HEX_METRIC_MAX_SHAPE              real_value
HEX_METRIC_MIN_RELATIVE_SIZE      real_value
HEX_METRIC_MAX_RELATIVE_SIZE      real_value
HEX_METRIC_MIN_SHAPE_AND_SIZE     real_value
HEX_METRIC_MAX_SHAPE_AND_SIZE     real_value

# Tet Metrics failure criteria:
TET_METRIC_MIN_ASPECT             real_value
TET_METRIC_MAX_ASPECT             real_value
TET_METRIC_MIN_ASPECT_GAMMA       real_value
TET_METRIC_MAX_ASPECT_GAMMA       real_value
TET_METRIC_MIN_VOLUME             real_value
TET_METRIC_MAX_VOLUME             real_value
TET_METRIC_MIN_CONDITION          real_value
TET_METRIC_MAX_CONDITION          real_value
TET_METRIC_MIN_JACOBIAN           real_value
TET_METRIC_MAX_JACOBIAN           real_value
```

TET_METRIC_MIN_NORM_JACOBIAN	real_value
TET_METRIC_MAX_NORM_JACOBIAN	real_value
TET_METRIC_MIN_SHEAR	real_value
TET_METRIC_MAX_SHEAR	real_value
TET_METRIC_MIN_SHAPE	real_value
TET_METRIC_MAX_SHAPE	real_value
TET_METRIC_MIN_RELATIVE_SIZE	real_value
TET_METRIC_MAX_RELATIVE_SIZE	real_value
TET_METRIC_MIN_SHAPE_AND_SIZE	real_value
TET_METRIC_MAX_SHAPE_AND_SIZE	real_value

Pyramid Metrics failure criteria:

PYRAMID_METRIC_MIN_VOLUME	real_value
PYRAMID_METRIC_MAX_VOLUME	real_value

Wedge Metrics failure criteria:

WEDGE_METRIC_MIN_VOLUME	real_value
WEDGE_METRIC_MAX_VOLUME	real_value

Knife Metrics failure criteria:

KNIFE_METRIC_MIN_VOLUME	real_value
KNIFE_METRIC_MAX_VOLUME	real_value

Quad Metrics failure criteria:

QUAD_METRIC_MIN_ASPECT	real_value
QUAD_METRIC_MAX_ASPECT	real_value
QUAD_METRIC_MIN_SKEW	real_value
QUAD_METRIC_MAX_SKEW	real_value
QUAD_METRIC_MIN_TAPER	real_value
QUAD_METRIC_MAX_TAPER	real_value
QUAD_METRIC_MIN_WARPAGE	real_value
QUAD_METRIC_MAX_WARPAGE	real_value
QUAD_METRIC_MIN_AREA	real_value
QUAD_METRIC_MAX_AREA	real_value
QUAD_METRIC_MIN_STRETCH	real_value
QUAD_METRIC_MAX_STRETCH	real_value
QUAD_METRIC_MIN_ANGLE	real_value
QUAD_METRIC_MAX_ANGLE	real_value
QUAD_METRIC_MIN_MIN_ANGLE	real_value
QUAD_METRIC_MAX_MIN_ANGLE	real_value
QUAD_METRIC_MIN_ODDY	real_value
QUAD_METRIC_MAX_ODDY	real_value
QUAD_METRIC_MIN_CONDITION	real_value
QUAD_METRIC_MAX_CONDITION	real_value
QUAD_METRIC_MIN_JACOBIAN	real_value
QUAD_METRIC_MAX_JACOBIAN	real_value
QUAD_METRIC_MIN_NORM_JACOBIAN	real_value
QUAD_METRIC_MAX_NORM_JACOBIAN	real_value
QUAD_METRIC_MIN_SHEAR	real_value
QUAD_METRIC_MAX_SHEAR	real_value
QUAD_METRIC_MIN_SHAPE	real_value

QUAD_METRIC_MAX_SHAPE	real_value
QUAD_METRIC_MIN_RELATIVE_SIZE	real_value
QUAD_METRIC_MAX_RELATIVE_SIZE	real_value
QUAD_METRIC_MIN_SHAPE_AND_SIZE	real_value
QUAD_METRIC_MAX_SHAPE_AND_SIZE	real_value

Tri Metrics failure criteria:

TRI_METRIC_MIN_ASPECT	real_value
TRI_METRIC_MAX_ASPECT	real_value
TRI_METRIC_MIN_AREA	real_value
TRI_METRIC_MAX_AREA	real_value
TRI_METRIC_MIN_ANGLE	real_value
TRI_METRIC_MAX_ANGLE	real_value
TRI_METRIC_MIN_CONDITION	real_value
TRI_METRIC_MAX_CONDITION	real_value
TRI_METRIC_MIN_NORM_JACOBIAN	real_value
TRI_METRIC_MAX_NORM_JACOBIAN	real_value
TRI_METRIC_MIN_SHEAR	real_value
TRI_METRIC_MAX_SHEAR	real_value
TRI_METRIC_MIN_SHAPE	real_value
TRI_METRIC_MAX_SHAPE	real_value
TRI_METRIC_MIN_RELATIVE_SIZE	real_value
TRI_METRIC_MAX_RELATIVE_SIZE	real_value
TRI_METRIC_MIN_SHAPE_AND_SIZE	real_value
TRI_METRIC_MAX_SHAPE_AND_SIZE	real_value

Edge Metrics failure criteria:

EDGE_METRIC_MIN_LENGTH	real_value
EDGE_METRIC_MAX_LENGTH	real_value

Appendix B

Descriptions of Verde Metrics

The following tables provide a description of the metrics that are applicable to each element type. Table B.1 lists the metrics that are applicable to hexahedral meshes.

Function Name	Dimension	Full Range	Acceptable Range	Reference
Aspect Ratio	L^0	1 to inf	1 to 4	[5]
Skew	L^0	0 to 1	0 to 0.5	[5]
Taper	L^0	0 to +inf	0 to 0.4	[5]
Element Volume	L^3	-inf to +inf	None	[5]
Stretch	L^0	0 to 1	0.25 to 1	[6]
Diagonal Ratio	L^0	0 to 1	0.65 to 1	
Dimension	L^1	0 to inf	None	[5]
Oddy	L^0	0 to inf	0 to 20	[7][8]
Condition No.	L^0	1 to inf	1 to 8	[8]
Jacobian	L^3	-inf to inf	None	[8]
Scaled Jacobian	L^0	-1 to +1	0.5 to 1	[8]
Shear	L^0	0 to 1	0.3 to 1	[9]
Shape	L^0	0 to 1	0.3 to 1	[9]
Relative Size	L^0	0 to 1	0.5 to 1	[9]
Shape & Size	L^0	0 to 1	0.2 to 1	[9]

Table B.1: Description of Hexahedral Quality Measures

The metrics for hexahedral elements as listed in table B.1 are defined as follows:

Aspect Ratio:	Maximum edge length ratios at hex center.
Skew:	Maximum $\|\cos A\|$ where A is the angle between edges at hex center.
Taper:	Maximum ratio of lengths derived from opposite edges.
Element Volume:	Jacobian at hex center.
Stretch:	$\sqrt{3}$ * minimum edge length / maximum diagonal length.
Diagonal Ratio:	Minimum diagonal length / maximum diagonal length.
Dimension:	Pronto-specific characteristic length for stable time step calculation. Char_length = Volume / 2 grad Volume.
Oddy:	General distortion measure based on left Cauchy-Green Tensor.
Condition No.	Maximum condition number of the Jacobian matrix at 8 corners.
Jacobian:	Minimum pointwise volume of local map at 8 corners & center of hex.
Scaled Jacobian:	Minimum Jacobian divided by the lengths of the 3 edge vectors.
Shear:	3/Condition number of Jacobian Skew matrix
Shape:	3/Condition number of weighted Jacobian matrix
Relative Size:	Min(J, 1/J), where J is determinant of weighted Jacobian matrix
Shape and Size:	Product of Shape and Relative Size

Table B.2 lists the metrics that are applicable to quadrilateral meshes.

Function Name	Dimension	Full Range	Acceptable Range	Reference
Aspect Ratio	L^0	1 to inf	1 to 4	[10]
Skew	L^0	0 to 1	0 to 0.5	[10]
Taper	L^0	0 to inf	0 to 0.7	[10]
Warpage	L^0	0 to inf	0 to 0.1	[10]
Element Area	L^2	-inf to +inf	None	[10]
Stretch	L^0	0 to 1	0.25 to 1	
Minimum Angle	degrees	0 to 90	45 to 90	
Maximum Angle	degrees	90 to 360	90 to 135	
Oddy	L^0	0 to inf	0 to 16	[7][8]
Condition No.	L^0	1 to inf	1 to 4	[8]
Jacobian	L^2	-inf to inf	None	[8]
Scaled Jacobian	L^0	-1 to +1	0.5 to 1	[8]
Shear	L^0	0 to 1	0.3 to 1	[9]
Shape	L^0	0 to 1	0.3 to 1	[9]
Relative Size	L^0	0 to 1	0.5 to 1	[9]
Shape & Size	L^0	0 to 1	0.2 to 1	[9]

Table B.2: Description of Quadrilateral Quality Measures

The metrics for quadrilateral elements as listed in table B.2 are defined as follows:

Aspect Ratio:	Maximum edge length ratios at quad center
Skew:	Maximum $\|\cos A\|$ where A is the angle between edges at quad center
Taper:	Maximum ratio of lengths derived from opposite edges
Warpage:	Deviation of element from planarity.
Element Area:	Jacobian at quad center
Stretch:	$\sqrt{2}$ * minimum edge length / maximum diagonal length
Minimum Angle:	Smallest included quad angle (degrees).
Maximum Angle:	Largest included quad angle (degrees).
Oddy:	General distortion measure based on left Cauchy-Green Tensor
Condition No.	Maximum condition number of the Jacobian matrix at 4 corners
Jacobian:	Minimum pointwise volume of local map at 4 corners & center of quad
Scaled Jacobian:	Minimum Jacobian divided by the lengths of the 2 edge vectors
Shear:	2/Condition number of Jacobian Skew matrix
Shape:	2/Condition number of weighted Jacobian matrix
Relative Size:	Min(J, 1/J), where J is determinant of weighted Jacobian matrix
Shape and Size:	Product of Shape and Relative Size

Table B.3 lists the metrics that are applicable to tetrahedral meshes.

Function Name	Dimension	Full Range	Acceptable Range	Reference
Aspect Ratio Bet	L^0	1 to inf	1 to 3	[11]
Aspect Ratio Gam	L^0	1 to inf	1 to 3	[11]
Element Volume	L^3	-inf to +inf	None	[11]
Condition No.	L^0	1 to inf	1 to 3	[8]
Jacobian	L^3	-inf to inf	None	[8]
Scaled Jacobian	L^0	-1.414 to +1.414	0.5 to +1.414	[8]
Shear	L^0	0 to 1	0.3 to 1	[9]
Shape	L^0	0 to 1	0.3 to 1	[9]
Relative Size	L^0	0 to 1	0.5 to 1	[9]
Shape & Size	L^0	0 to 1	0.2 to 1	[9]

Table B.3: Description of Tetrahedral Quality Measures

The metrics for tetrahedral elements as listed in table B.3 are defined as follows:

Aspect Ratio Beta:	CR / (3.0 * IR) where CR = circumsphere radius, IR = inscribed sphere radius
Aspect Ratio Gamma:	$Srms^3/(8.479670 * V)$ where $Srms = \sqrt{Sum(Si^2)/6}$, Si = edge length
Element Volume:	(1/6) * Jacobian at corner node
Condition No.	Condition number of the Jacobian matrix at any corner
Jacobian:	Minimum pointwise volume at any corner
Scaled Jacobian:	Minimum Jacobian divided by the lengths of 3 edge vectors
Shear:	3/Minimum condition number of weighted Skew matrices
Shape:	3/Condition number of weighted Jacobian matrix
Relative Size:	Min(J, 1/J), where J is determinant of weighted Jacobian matrix
Shape and Size:	Product of Shape and Relative Size

Table B.4 lists the metrics that are applicable to triangular meshes.

Function Name	Dimension	Full Range	Acceptable Range	Reference
Aspect Ratio Gam	L^0	1 to inf	1 to 1.3	[11]
Element Area	L^2	0 to inf	None	
Maximum Angle	degrees	60 to 180	60 to 90	
Minimum Angle	degrees	0 to 60	30 to 60	
Condition No.	L^0	1 to inf	1 to 1.3	[8]
Scaled Jacobian	L^0	-1.155 to +1.155	0.5 to 1.155	[8]
Shear	L^0	0 to 1	0.75 to 1	[9]
Shape	L^0	0 to 1	0.75 to 1	[9]
Relative Size	L^0	0 to 1	0.75 to 1	[9]
Shape & Size	L^0	0 to 1	0.75 to 1	[9]

Table B.4: Description of Triangular Quality Measures

The metrics for triangular elements as listed in table B.4 are defined as follows:

Aspect Ratio Gamma:	$srms^2/(2.30940108 * area)$ where $srms = \sqrt{\sum Si^2/3}$ and $Si =$ edge length
Element Area:	(1/2) * Jacobian at corner node
Maximum Angle:	Maximum included angle in triangle
Minimum Angle:	Minimum included angle in triangle
Condition No.	Condition number of the Jacobian matrix at any corner
Scaled Jacobian:	Minimum Jacobian divided by the lengths of 2 edge vectors
Shear:	2/Condition number of weighted skew matrix
Shape:	2/Condition number of weighted Jacobian matrix
Relative Size:	Min(J, 1/J), where J is determinant of weighted
Jacobian matrix	
Shape and Size:	Product of Shape and Relative Size

Appendix C

Command Line and Journal File Keywords

The following table list all the text based commands that can be used to control Verde from either the command line or a journal file. Subsequent columns list the options that are applicable to each major keyword. This information is also available by typing “help” at the command line prompt.

Table C.1: Text based commands

Major Command	**Option 1**	**Option 2**	**Option 3**
block	$< id >$ (integer)	on off	
calculate			
display			
echo	on off		
elementbc	$< id >$ (integer)	on off	
exit			
graphics	mode	wireframe hiddenline smoothshade	
	reset		
help			
highlight failed	hexes tets quads tris	metric	[metric_name]
load	defaults mesh	'$< filename >$'	
nodeset	$< id >$ (integer)	on off	
normals	on off		
playback	'$< filename >$'		

Table C.1: continued

Major Command	Option 1	Option 2	Option 3
quit			
record	'$< journal\_filename >$' stop		
set	topology_calculations	on off	
	interface_calculations	on off	
	metric_calculations	on off	
	verbose_model	on off	
	individual_output	on off	
	warning level	on off	
	hex	aspect skew taper volume stretch diagonal dimension oddy condition jacobian scaled jacobian shear shape relative Size shape and Size	maximum (value) minimum (value)
	tet	aspect aspect gamma volume condition jacobian scaled jacobian shear shape relative size shape and size	maximum (value) minimum (value)
	pyramid	volume	maximum (value) minimum (value)
	wedge	volume	maximum (value) minimum (value)
	knife	volume	maximum (value) minimum (value)
	quad	aspect skew	maximum (value) minimum (value)

Table C.1: continued

Major Command	Option 1	Option 2	Option 3
		taper warpage area stretch smallest angle largest angle oddy condition jacobian scaled jacobian shear shape relative size shape and size	
	tri	aspect area largest angle smallest angle condition scaled jacobian shear shape relative size shape and size	maximum (value) minimum (value)
	edge	length	maximum (value) minimum (value)

Bibliography

[1] L. Schoof and V. Yarberry, “Exodus II: A finite element data model,” Tech. Rep. SAND92-2137, Sandia National Laboritories, Albuqurque, NM, 1996.

[2] R. Rew, G. Davis, S. Emmerson, and H. Davies, *NetCDF User's Guide for C, An Interface for Data Access*, 3rd ed., Apr. 1997.

[3] http://www.trolltech.com, 2001.

[4] W. Beyer, ed., *CRC Standard Mathematical Tables.* Boca Raton, FL: CRC Press, Inc., 27 ed., 1985.

[5] L. Taylor and D. Flanagan, “Pronto3D - A three dimensional transient solid dynamics program,” tech. rep., Sandia National Laboritories, Albuqurque, NM, 1989.

[6] *FIMESH Code.*

[7] A. Oddy, J. Goldak, M. McDill, and M. Bibby, “A distortion metric for isoparametric finite elements,” *Trans. CSME*, vol. 38-CSME-32, no. Accession No. 2161, 1988.

[8] P. Knupp, “Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities, Part II - A framework for volume mesh optimization and the condition number of the Jacobian matrix,” *Int. J. Numer. Meth. Engng.*, no. 48, pp. 1165–1185., 2000.

[9] P. M. Knupp, “Algebraic mesh quality metrics,” *SIAM Journal on Scientific Computing*, vol. 23, pp. 193–218, Jan. 2002.

[10] J. Robinson, “CRE method of element testing and the jacobian shape parameters,” *Eng. Comput.*, vol. 4, pp. 113–118, 1987.

[11] V. Parthasarathy, C. Graichen, and A. Hathaway, “A comparison of tetrahedron quality measures,” *Finite Elements in Analysis and Design*, vol. 15, pp. 225–261, 1993.